

		Committed Backlog Items	Tasks	Who?	Not Started	In Progress	Completed
<b>PBI #1</b>		<b>PBI #1 - Generate error messages</b>	<b>PBI time: 8hr</b>	<b>Alex</b>		x	
As	an election official using the voting system software	0.5hr	Catch error conditions	Alex			x
I want	the program to provide appropriate error messages if a bad ballot file is passed in	0.5hr	Generate useful error messages	Alex			x
So that	I can fix the file and provide the correctly formatted file	0.5hr	Clearly resolve error	Alex			x
		4hr	Fuzz via AFL++	Alex			x
Acceptance Criteria	Generate appropriate errors for empty ballot files, files with parse problems	0.5hr	Documentation	Alex			x
Definition of Done	Generate an error if unable to parse input CSV file; accept correctly formatted CSV files; No compilation error; unit tests for each method and agreed amount of system tests pass; correct output	1hr	Write new unit tests	Alex			x
Effort	Medium (~5-6 hours)	0.1hr	Run new unit tests	Alex			x
		0.5hr	Write new system tests	Alex			x
		0.1hr	Run new system tests	Alex			x
		0.15hr	Write unit test logs	Alex	x		
		0.15hr	Write system test logs	Alex	x		
<b>PBI #2</b>		<b>PBI #2 - Resolve ties via coin flip</b>	<b>PBI time: 3.5hr</b>	<b>Alex</b>		x	
As	an election official using the voting system software	1hr	Add new functionality	Alex			x
I want	Candidates or parties with ties to be resolved via a 50/50 coin flip	0.5hr	Documentation	Alex			x
So that	Ties are resolved randomly, fairly, and independently of other factors	1hr	Write new unit tests	Alex	x		
		0.1hr	Run new unit tests	Alex	x		
Acceptance Criteria	Ties between candidates or between parties can be resolved randomly	0.5hr	Write new system tests	Alex	x		
Definition of Done	Ties between candidates or between parties can be resolved randomly; No compilation error; unit tests for each method and agreed amount of system tests pass; correct output	0.1hr	Run new system tests	Alex	x		
Effort	Small (~2-3 hours)	0.15hr	Write unit test logs	Alex	x		
		0.15hr	Write system test logs	Alex	x		
<b>PBI #3</b>		<b>PBI #3 - VotingSystem api has one method for elections</b>	<b>PBI time: 2.6hr</b>	<b>Alex</b>		x	
As	a developer	0.1hr	Rewrite old calls in Election	Alex			x
I want	to be able to create and run an election, easily. The electables field is not populated unless three additional functions are called in order	0.5hr	Documentation	Alex			x
So that	The probability of logic errors are reduced and the system is easily extensible.	1hr	Write new unit tests	Alex	x		
		0.1hr	Run new unit tests	Alex	x		
Acceptance Criteria	A new method is added to votingSystem to automatically call processElectables(), countVotes(), and calculateResults().	0.5hr	Write new system tests	Alex			x
Definition of Done	No compilation error; unit tests for each method and agreed amount of system tests pass; correct output	0.1hr	Run new system tests	Alex			x
Effort	Small (~2-3 hours)	0.15hr	Write unit test logs	Alex			x
		0.15hr	Write system test logs	Alex			x
<b>PBI #4</b>		<b>PBI #4 - Multiple files for the CPL algorithm</b>	<b>PBI time: 3.5hr</b>	<b>Ashwin, Alex, Leo</b>		x	
As	an election official	0.25hr	Change CLI functionality to take in multiple files for CPL	Alex			x
I want	Multiple files to be brought in for a CPL election	0.25hr	Change GUI functionality to take in multiple files for CPL	Alex			x
So that	they can bring in different files from different balloting locations	1hr	Code functionality for multiple files read	Leo		x	x
		0.5hr	Write new unit tests for new methods for CPL multiple file functionality	Ashwin	x		
Acceptance Criteria	Multiple .csv input files for CPL election being read correctly at the same time	0.1hr	Run new unit tests for new methods written	Ashwin	x		
Definition of Done	No compilation error; unit tests for each method and agreed amount of system tests pass; correct output	0.5hr	Write integration tests for CPL multiple files	Leo		x	
Effort	Small (~2-3 hours)	0.1hr	Run integration tests to make sure system still functions correctly	Leo		x	
		0.3hr	Write test logs for unit and integration tests	Leo		x	
		0.5hr	Write Doxygen documentation for new methods written	Ashwin	x		
<b>PBI #5</b>		<b>PBI #5 - Multiple files for the OPL algorithm</b>	<b>PBI time: 3.5hr</b>	<b>Ashwin, Leo</b>		x	
As	an election official	0.25hr	Change CLI functionality to take in multiple files for OPL	Ashwin			x
I want	Multiple files to be brought in for an OPL election	0.25hr	Change GUI functionality to take in multiple files for OPL	Ashwin			x
So that	they can bring in different files from different balloting locations	1hr	Code functionality for multiple files read	Leo			x
		0.5hr	Write new unit tests for new methods for OPL multiple file functionality	Leo		x	
Acceptance Criteria	Multiple .csv input files for OPL election being read correctly at the same time	0.1hr	Run new unit tests for new methods written	Leo		x	
Definition of Done	No compilation error; unit tests for each method and agreed amount of system tests pass; correct output	0.5hr	Write integration tests for OPL multiple files	Leo	x		
Effort	Small (~2-3 hours)	0.1hr	Run integration tests to make sure system still functions correctly	Leo	x		
		0.3hr	Write test logs for unit and integration tests	Ashwin	x		
		0.5hr	Write Doxygen documentation for new methods written	Leo	x		
<b>PBI #6</b>		<b>PBI #6 - MPO files in general</b>	<b>PBI time: 3hr</b>	<b>Janani, Leo</b>			
As	an election official	0.5hr	MPO can read MPO file	Janani		x	
I want	MPO election ballots to be brought in via a .csv file	0.5hr	MPO correctly inherits from VotingSystem and implements all required methods	Janani		x	
So that	the election can be run by the MPO algorithm in the system	0.5hr	Write unit tests for added functionality	Janani		x	
		0.1hr	Run unit tests	Janani		x	
Acceptance Criteria	.csv input file for MPO election being read correctly	0.5hr	Write integration tests to make sure the system still works as intended	Leo		x	

Definition of Done	No compilation error; unit tests for each method and agreed amount of system tests pass; correct output	0.1hr	Run integration tests	Leo	x		
Effort	Small (~2-3 hours)	0.15hr	Write unit test logs	Leo	x		
		0.15hr	Write system test logs	Leo	x		
		0.5hr	Write Doxygen documentation for new methods written	Leo	x		
<b>PBI #7</b>		<b>PBI #7 - MPO multiple files</b>	<b>PBI time: 2.7hr</b>	<b>Alex, Ashwin, Leo</b>	x		
As	an election official	0.1hr	Change CLI functionality to take in multiple files for MPO	Alex			x
I want	Multiple files to be brought in for a MPO election	0.1hr	Change GUI functionality to take in multiple files for MPO	Alex			x
So that	they can bring in different files from different balloting locations	0.5hr	Code functionality for multiple files read	Leo	x		x
		0.5hr	Write new unit tests for new methods for MPO multiple file functionality	Ashwin	x		
Acceptance Criteria	Multiple .csv input files for MPO election being read correctly at the same time	0.1hr	Run new unit tests for new methods written	Ashwin	x		
Definition of Done	No compilation error; unit tests for each method and agreed amount of system tests pass; correct output	0.5hr	Write integration tests for MPO multiple files	Ashwin	x		
Effort	Small (~2-3 hours)	0.1hr	Run Integration tests to make sure system still functions correctly	Ashwin	x		
		0.3hr	Write test logs for unit and integration tests	Leo	x		
		0.5hr	Write Doxygen documentation for new methods written	Leo	x		
<b>PBI #8</b>		<b>PBI #8 - MPO election</b>	<b>PBI time: 4.5hr</b>	<b>Alex, Leo, Ashwin</b>	x		
As	an election official	1hr	Break RawData::ParseFile() into sub functions	Alex			x
I want	to write an algorithm that determines multiple winners based on the candidates receiving the most ballots with a fair coin toss	0.5hr	Change header line read order based on election type	Alex			x
So that	an MPO election can be run by the voting system	1hr	Add new functionality	Leo	x		
		0.5hr	Write unit tests for added functionality	Leo	x		
Acceptance Criteria	MPO algorithm correctly calculates the winners and losers and fairly determines ties if there are any	0.1hr	Run unit tests	Leo	x		
Definition of Done	No compilation error; unit tests for each method and agreed amount of system tests pass; correct and fair output	0.5hr	Write integration tests to make sure the system still works as intended	Leo	x		
Effort	Large (~6-7 hours)	0.1hr	Run integration tests	Leo	x		
		0.15hr	Write unit test logs	Ashwin			
		0.15hr	Write system test logs	Ashwin			
		0.5hr	Write Doxygen documentation for new methods written	Ashwin			
<b>PBI #9</b>		<b>PBI #9 - MV files in general</b>	<b>PBI time: 3hr</b>	<b>Ashwin, Janani</b>			
As	an election official	0.5hr	RawData can read MV file	Ashwin	x		
I want	MV election ballots to be brought in via a .csv file	0.5hr	Fileops can read MV file	Ashwin	x		
So that	the election can be run by the MV algorithm in the system	0.5hr	Write unit tests for added functionality	Ashwin	x		
		0.1hr	Run unit tests	Ashwin	x		
Acceptance Criteria	.csv input file for MV election being read correctly	0.5hr	Write integration tests to make sure the system still works as intended	Janani	x		
Definition of Done	No compilation error; unit tests for each method and agreed amount of system tests pass; correct output	0.1hr	Run integration tests	Janani	x		
Effort	Small (~2-3 hours)	0.15hr	Write unit test logs	Janani	x		
		0.15hr	Write system test logs	Janani	x		
		0.5hr	Write Doxygen documentation for new methods written	Ashwin	x		
<b>PBI #10</b>		<b>PBI #10 - MV multiple files</b>	<b>PBI time: 2.7hr</b>	<b>Alex, Leo</b>	x		
As	an election official	0.1hr	Change CLI functionality to take in multiple files for MV	Alex			x
I want	Multiple files to be brought in for an MV election	0.1hr	Change GUI functionality to take in multiple files for MV	Alex			x
So that	Different files can be brought in from different balloting locations	0.5hr	Code functionality for multiple files read	Leo			x
		0.5hr	Write new unit tests for new methods for MV multiple file functionality	Leo	x		
Acceptance Criteria	Multiple .csv input files for MV election being read correctly at the same time	0.1hr	Run new unit tests for new methods written	Leo	x		
Definition of Done	No compilation error; unit tests for each method and agreed amount of system tests pass; correct output	0.5hr	Write integration tests for MV multiple files	Leo	x		
Effort	Small (~2-3 hours)	0.1hr	Run Integration tests to make sure system still functions correctly	Leo	x		
		0.3hr	Write test logs for unit and integration tests	Leo	x		
		0.5hr	Write Doxygen documentation for new methods written	Leo	x		
			<b>Total time: 37hr</b>				